Package ‘mapr’

July 30, 2020

Title Visualize Species Occurrence Data

Description Utilities for visualizing species occurrence data. Includes functions to visualize occurrence data from 'spocc', 'rgbif', and other packages. Mapping options included for base R plots, 'ggplot2', 'ggmap', 'leaflet' and 'GitHub' 'gists'.

Version 0.5.0

License MIT + file LICENSE

URL `https://docs.ropensci.org/mapr`, `https://github.com/ropensci/mapr`

BugReports `https://github.com/ropensci/mapr/issues`

LazyData true

LazyLoad true

VignetteBuilder knitr

Encoding UTF-8

Language en-US

Imports ggplot2, leaflet, spocc (>= 0.6.0), sp, maps, RColorBrewer, jsonlite, gistr, data.table

Suggests testthat, knitr, taxize, maptools, rgbif

RoxygenNote 7.1.1

X-schema.org-applicationCategory Geospatial

X-schema.org-keywords maps, mapping, ggplot, leaflet, interactive, biodiversity, occurrences

X-schema.org-isPartOf https://ropensci.org

NeedsCompilation no

Author Scott Chamberlain [aut, cre] (<https://orcid.org/0000-0003-1444-9135>)

Maintainer Scott Chamberlain <`myrmecocystus@gmail.com`>

Repository CRAN

Date/Publication 2020-07-30 04:20:07 UTC

R topics documented:

`mapr-package` *mapr*

Description

Visualize species occurrence data

Many inputs

All functions take the following kinds of inputs:

- An object of class `occdat`, from the package **spocc**. An object of this class is composed of many objects of class `occdatind`
- An object of class `occdatind`, from the package **spocc**
- An object of class `gbif`, from the package **rgbif**
- An object of class `data.frame`. This data.frame can have any columns, but must include a column for taxonomic names (e.g., `name`), and for latitude and longitude (we guess your lat/long columns, starting with the default `latitude` and `longitude`)
- An object of class `SpatialPoints`
- An object of class `SpatialPointsDatFrame`

Package API

- `map_plot()` - static Base R plots
- `map_ggplot()` - static ggplot2 plots
- `map_leaflet()` - interactive Leaflet.js interactive maps
- `map_gist()` - ineractive, shareable maps on GitHub Gists

Author(s)

Scott Chamberlain <`myrmecocystus@gmail.com`>

gbif_eg1 — *Example dataset: output from call to* `rgbif::occ_search()`

Description

A dataset with 50 rows, and 101 columns, from the query: `rgbif::occ_search(scientificName = "Puma concolor",limit = 100)`

Format

A data frame with 50 rows and 101 variables

Details

See `inst/ignore/datasets.R` for the code to prepare this dataaset

hull — *Add a convex hull to a map*

Description

Add a convex hull to a map

Usage

```
hull(x, ...)
```

Arguments

`x`	input
`...`	ignored

Details

Can be used with `map_leaflet()`, `map_plot()`, and `map_ggplot()`. Other methods in this package may be supported in the future.

Value

Adds a convex hull to the plot. See `grDevices::chull()` for info.

Examples

```
# map spocc output, here using a built in object
data(occdat_eg1)
map_plot(occdat_eg1, hull = TRUE)

# map rgbif output, here using a built in object
hull(map_ggplot(occdat_eg1))

## Not run:
# leaflet
library("spocc")
spp <- c('Danaus plexippus', 'Accipiter striatus', 'Pinus contorta')
dat <- occ(spp, from = 'gbif', limit = 30, has_coords = TRUE)
hull(map_leaflet(dat))

# ggplot
if (requireNamespace("rgbif")) {
library("rgbif")
res <- occ_search(scientificName = "Puma concolor", limit = 100)
hull(map_ggplot(res))
}

# base plots
library("spocc")
out <- occ(query='Accipiter striatus', from='gbif', limit=25,
  has_coords=TRUE)
map_plot(out, hull = TRUE)

## End(Not run)
```

map_ggplot *ggplot2 mapping*

Description

ggplot2 mapping

Usage

```
map_ggplot(
  x,
  map = "world",
  point_color = "#86161f",
  color = NULL,
  size = 3,
  lon = "longitude",
  lat = "latitude",
  name = NULL,
  ...
)
```

Arguments

`x`	The data. An object of class `occdat`, `occdatind`, `gbif`, `gbif_data`, `SpatialPoints`, `SpatialPointsDataFrame`, or `data.frame`. The package **spocc** needed for the first two, and **rgbif** needed for the third. When `data.frame` input, any number of columns allowed, but with at least the following: name (the taxonomic name), latitude (in dec. deg.), longitude (in dec. deg.)
`map`	(character) One of world, world2, state, usa, county, france, italy, or nz
`point_color`	Default color of your points. Deprecated, use `color`
`color`	Default color of your points.
`size`	point size, Default: 3
`lon, lat`	(character) Longitude and latitude variable names. Ignored unless `data.frame` input to `x` parameter. We attempt to guess, but if nothing close, we stop. Default: `longitude` and `latitude`
`name`	(character) the column name that contains the name to use in creating the map. If left `NULL` we look for a "name" column. default: `NULL`
`...`	Ignored

Value

A ggplot2 map, of class `gg/ggplot`

Examples

```
# map spocc output, here using a built in object
data(occdat_eg1)
map_ggplot(occdat_eg1)

# map rgbif output, here using a built in object
data(gbif_eg1)
map_ggplot(gbif_eg1)

## Not run:
## spocc
library("spocc")
ddat <- occ(query = 'Lynx rufus californicus', from = 'gbif', limit=100)
map_ggplot(ddat)
map_ggplot(ddat$gbif)
map_ggplot(ddat$gbif, "usa")
map_ggplot(ddat, "county")

### usage of occ2sp()
#### SpatialPoints
spdat <- occ2sp(ddat)
map_ggplot(spdat)
#### SpatialPointsDataFrame
spdatdf <- as(spdat, "SpatialPointsDataFrame")
map_ggplot(spdatdf)
```

```
## rgbif
if (requireNamespace("rgbif")) {
library("rgbif")
library("ggplot2")
### occ_search() output
res <- occ_search(scientificName = "Puma concolor", limit = 100)
map_ggplot(res)

### occ_data() output
res <- occ_data(scientificName = "Puma concolor", limit = 100)
map_ggplot(res)

#### many taxa
res <- occ_data(scientificName = c("Puma concolor", "Quercus lobata"),
   limit = 30)
map_ggplot(res)

### add a convex hull
hull(map_ggplot(res))
}

## data.frame
df <- data.frame(name = c('Poa annua', 'Puma concolor', 'Foo bar'),
                 longitude = c(-120, -121, -121),
                 latitude = c(41, 42, 45), stringsAsFactors = FALSE)
map_ggplot(df)

# many species, each gets a different color
library("spocc")
spp <- c('Danaus plexippus', 'Accipiter striatus', 'Pinus contorta')
dat <- occ(spp, from = 'gbif', limit = 30, has_coords = TRUE)
map_ggplot(dat, color = c('#976AAE', '#6B944D', '#BD5945'))

## End(Not run)
```

map_gist *Make an interactive map to view in the browser as a GitHub gist*

Description

Make an interactive map to view in the browser as a GitHub gist

Usage

```
map_gist(
  x,
  description = "",
  public = TRUE,
  browse = TRUE,
  lon = "longitude",
  lat = "latitude",
  name = NULL,
  ...
)
```

Arguments

`x`	The data. An object of class `occdat`, `occdatind`, `gbif`, `gbif_data`, `SpatialPoints`, `SpatialPointsDataFrame`, or `data.frame`. The package **spocc** needed for the first two, and **rgbif** needed for the third. When `data.frame` input, any number of columns allowed, but with at least the following: name (the taxonomic name), latitude (in dec. deg.), longitude (in dec. deg.)
`description`	Description for the Github gist, or leave to default (=no description)
`public`	(logical) Whether gist is public (default: `TRUE`)
`browse`	If `TRUE` (default) the map opens in your default browser.
`lon, lat`	(character) Longitude and latitude variable names. Ignored unless `data.frame` input to `x` parameter. We attempt to guess, but if nothing close, we stop. Default: `longitude` and `latitude`
`name`	(character) the column name that contains the name to use in creating the map. If left `NULL` we look for a "name" column.
`...`	Further arguments passed on to `style_geojson()`

Details

See `gistr::gist_auth()` for help on authentication

Does not support adding a convex hull via `hull()`

Examples

```
## Not run:
## spocc
library("spocc")
spp <- c('Danaus plexippus', 'Accipiter striatus', 'Pinus contorta')
dat <- occ(spp, from=c('gbif','ecoengine'), limit=30,
  gbifopts=list(hasCoordinate=TRUE))
dat <- fixnames(dat, "query")

# Define colors
map_gist(dat, color=c('#976AAE','#6B944D','#BD5945'))
map_gist(dat$gbif, color=c('#976AAE','#6B944D','#BD5945'))
map_gist(dat$ecoengine, color=c('#976AAE','#6B944D','#BD5945'))

# Define colors and marker size
map_gist(dat, color=c('#976AAE','#6B944D','#BD5945'),
  size=c('small','medium','large'))

# Define symbols
map_gist(dat, symbol=c('park','zoo','garden'))
```

```
## rgbif
if (requireNamespace("rgbif")) {
library("rgbif")
### occ_search() output
res <- occ_search(scientificName = "Puma concolor", limit = 100)
map_gist(res)

### occ_data() output
res <- occ_data(scientificName = "Puma concolor", limit = 100)
map_gist(res)

#### many taxa
res <- occ_data(scientificName = c("Puma concolor", "Quercus lobata"),
   limit = 30)
res
map_gist(res)
}

## data.frame
df <- data.frame(name = c('Poa annua', 'Puma concolor', 'Foo bar'),
                 longitude = c(-120, -121, -121),
                 latitude = c(41, 42, 45), stringsAsFactors = FALSE)
map_gist(df)

### usage of occ2sp()
#### SpatialPoints
spdat <- occ2sp(dat)
map_gist(spdat)
#### SpatialPointsDataFrame
spdatdf <- as(spdat, "SpatialPointsDataFrame")
map_gist(spdatdf)

## End(Not run)
```

map_leaflet *Make interactive maps with Leaflet.js*

Description

Make interactive maps with Leaflet.js

Usage

```
map_leaflet(
  x,
  lon = "longitude",
  lat = "latitude",
  color = NULL,
  size = 13,
  name = NULL,
  ...
)
```

Arguments

`x`	The data. An object of class `occdat`, `occdatind`, `gbif`, `gbif_data`, `SpatialPoints`, `SpatialPointsDataFrame`, or `data.frame`. The package **spocc** needed for the first two, and **rgbif** needed for the third. When `data.frame` input, any number of columns allowed, but with at least the following: name (the taxonomic name), latitude (in dec. deg.), longitude (in dec. deg.)
`lon, lat`	(character) Longitude and latitude variable names. Ignored unless `data.frame` input to `x` parameter. We attempt to guess, but if nothing close, we stop. Default: `longitude` and `latitude`
`color`	Default color of your points.
`size`	point size, Default: 13
`name`	(character) the column name that contains the name to use in creating the map. If left `NULL` we look for a "name" column.
`...`	Ignored

Details

We add popups by default, and add all columns to the popup. The html is escaped with `htmltools::htmlEscape()`

Value

a Leaflet map in Viewer in Rstudio, or in your default browser otherwise

Examples

```
## Not run:
## spocc
library("spocc")
(out <- occ(query='Accipiter striatus', from='gbif', limit=50,
  has_coords=TRUE))
### with class occdat
map_leaflet(out)
### with class occdatind
map_leaflet(out$gbif)
### use occ2sp
map_leaflet(occ2sp(out))

## rgbif
if (requireNamespace("rgbif")) {
library("rgbif")
res <- occ_search(scientificName = "Puma concolor", limit = 100)
map_leaflet(res)
}

## SpatialPoints class
library("sp")
```

```
df <- data.frame(longitude = c(-120,-121),
                 latitude = c(41, 42), stringsAsFactors = FALSE)
x <- SpatialPoints(df)
map_leaflet(x)

## SpatialPointsDataFrame class
if (requireNamespace("rgbif")) {
library("rgbif")
### occ_search() output
res <- occ_search(scientificName = "Puma concolor", limit = 100)
x <- res$data
library("sp")
x <- x[stats::complete.cases(x$decimalLatitude, x$decimalLongitude), ]
coordinates(x) <- ~decimalLongitude+decimalLatitude
map_leaflet(x)

### occ_data() output
res <- occ_data(scientificName = "Puma concolor", limit = 100)
map_leaflet(res)
}

#### many taxa
res <- occ_data(scientificName = c("Puma concolor", "Quercus lobata"),
   limit = 30)
res
map_leaflet(res)


## data.frame
df <- data.frame(name = c('Poa annua', 'Puma concolor'),
                 longitude = c(-120,-121),
                 latitude = c(41, 42), stringsAsFactors = FALSE)
map_leaflet(df)

# many species
library("spocc")
spp <- c('Danaus plexippus', 'Accipiter striatus', 'Pinus contorta')
dat <- occ(spp, from = 'gbif', limit = 50, has_coords = TRUE)
map_leaflet(dat)
map_leaflet(dat, color = c('#AFFF71', '#AFFF71', '#AFFF71'))
map_leaflet(dat, color = c('#976AAE', '#6B944D', '#BD5945'))

# add a convex hull
## map_leaflet(dat) %>% hull()  # using pipes
hull(map_leaflet(dat))

## End(Not run)
```

map_plot *Base R mapping*

Description

Base R mapping

Usage

```
map_plot(
  x,
  lon = "longitude",
  lat = "latitude",
  color = NULL,
  size = 1,
  pch = 16,
  hull = FALSE,
  name = NULL,
  ...
)
```

Arguments

`x`	The data. An object of class `occdat`, `occdatind`, `gbif`, `gbif_data`, `SpatialPoints`, `SpatialPointsDataFrame`, or `data.frame`. The package **spocc** needed for the first two, and **rgbif** needed for the third. When `data.frame` input, any number of columns allowed, but with at least the following: name (the taxonomic name), latitude (in dec. deg.), longitude (in dec. deg.)
`lon, lat`	(character) Longitude and latitude variable names. Ignored unless `data.frame` input to `x` parameter. We attempt to guess, but if nothing close, we stop. Default: `longitude` and `latitude`
`color`	Default color of your points.
`size`	point size, passed to `cex` Default: 1
`pch`	point symbol shape, Default: 16
`hull`	(logical) whether to add a convex hull. Default: `FALSE`
`name`	(character) the column name that contains the name to use in creating the map. If left `NULL` we look for a "name" column.
`...`	Further args to `graphics::points()`

Value

Plots a world scale map

Examples

```
# map spocc output, here using a built in object
data(occdat_eg1)
map_plot(occdat_eg1)

# map rgbif output, here using a built in object
data(gbif_eg1)
```

```
map_plot(gbif_eg1)

## Not run:
## spocc
library("spocc")
(out <- occ(query='Accipiter striatus', from='gbif', limit=25,
  has_coords=TRUE))
### class occdat
map_plot(out)
map_plot(out, hull = TRUE)
### class occdatind
map_plot(out$gbif)
map_plot(out$gbif, hull = TRUE)

## rgbif
if (requireNamespace("rgbif")) {
library("rgbif")
### occ_search() output
res <- occ_search(scientificName = "Puma concolor", limit = 100)
map_plot(res)
map_plot(res, hull = TRUE)

### occ_data() output
res <- occ_data(scientificName = "Puma concolor", limit = 100)
map_plot(res)
#### many taxa
res <- occ_data(scientificName = c("Puma concolor", "Quercus lobata"),
   limit = 30)
res
map_plot(res)
}


## data.frame
df <- data.frame(
  name = c('Poa annua', 'Puma concolor', 'Foo bar', 'Stuff things'),
  longitude = c(-125, -123, -121, -110),
  latitude = c(41, 42, 45, 30), stringsAsFactors = FALSE)
map_plot(df)
map_plot(df, hull = TRUE)

### usage of occ2sp()
#### SpatialPoints
spdat <- occ2sp(out)
map_plot(spdat)
map_plot(spdat, hull = TRUE)

#### SpatialPointsDataFrame
spdatdf <- as(spdat, "SpatialPointsDataFrame")
map_plot(spdatdf)
map_plot(spdatdf, hull = TRUE)

# many species, each gets a different color
```

```
library("spocc")
spp <- c('Danaus plexippus', 'Accipiter striatus', 'Pinus contorta',
  'Ursus americanus')
dat <- occ(spp, from = 'gbif', limit = 30, has_coords = TRUE,
  gbifopts = list(country = 'US'))
map_plot(dat)
map_plot(dat, hull = TRUE)
## diff. color for each taxon
map_plot(dat, color = c('#976AAE', '#6B944D', '#BD5945', 'red'))
map_plot(dat, color = c('#976AAE', '#6B944D', '#BD5945', 'red'), hull = TRUE)

# add a convex hull
if (requireNamespace("rgbif")) {
library("rgbif")
res <- occ_search(scientificName = "Puma concolor", limit = 100)
map_plot(res, hull = FALSE)
map_plot(res, hull = TRUE)
}

## End(Not run)
```

occ2sp — *Create a spatial points dataframe from a spocc search*

Description

Create a spatial points dataframe from a spocc search

Usage

```
occ2sp(x, coord_string = "+proj=longlat +datum=WGS84", just_coords = FALSE)
```

Arguments

x	The resuslts of a spocc search called by `spocc::occ()`
coord_string	A valid EPGS cooridate string from the sp package, the default is WSGS 84
just_coords	Return data frame with specios names and provenance or just a spatial points object, which is the default.

Details

This function will return either a spatial points dataframe or spatial points object. Conversion to spatial points objects allows spocc searches to interact with other spatial data sources. More coordinate system codes can be found at the EPGS registry: http://www.epsg-registry.org/

Examples

```
## Not run:
### See points on a map
library("maptools")
library("spocc")
data(wrld_simpl)
plot(wrld_simpl[wrld_simpl$NAME == "United States", ], xlim = c(-70, -60))
out <- occ(query = "Accipiter striatus", from = c("vertnet", "gbif"),
  limit = 50)
xx <- occ2sp(out, just_coords = TRUE)
points(xx, col = 2)

## End(Not run)
```

occdat_eg1 — *Example dataset: output from call to* `spocc::occ()`

Description

A dataset with 25 rows, and 62 columns, from the query: `occ(query='Accipiter striatus',from='gbif',limit=25,has`

Format

A data frame with 25 rows and 62 variables

Details

See `inst/ignore/datasets.R` for the code to prepare this dataaset

style_geojson — *Style a data.frame prior to converting to geojson.*

Description

Style a data.frame prior to converting to geojson.

Usage

```
style_geojson(
  input,
  var = NULL,
  var_col = NULL,
  var_sym = NULL,
  var_size = NULL,
  color = NULL,
  symbol = NULL,
  size = NULL
)
```

Arguments

`input`	A data.frame
`var`	A single variable to map colors, symbols, and/or sizes to.
`var_col`	The variable to map colors to.
`var_sym`	The variable to map symbols to.
`var_size`	The variable to map size to.
`color`	Valid RGB hex color
`symbol`	An icon ID from the Maki project `http://www.mapbox.com/maki/` or a single alphanumeric character (a-z or 0-9).
`size`	One of 'small', 'medium', or 'large'

Index